

A Two-Step High-Order Compact Scheme for the Laplacian Operator and its Implementation in an Explicit Method for Integrating the Nonlinear Schrödinger Equation.

R. M. Caplan* and R. Carretero-González

Nonlinear Dynamical Systems Group[†], Computational Science Research Center[‡] and
Department of Mathematics and Statistics, San Diego State University,
San Diego, California 92182-7720, USA

September 7, 2011

Abstract

We describe and test an easy-to-implement two-step high-order compact (2SHOC) scheme for the Laplacian operator and its implementation into an explicit finite-difference scheme for simulating the nonlinear Schrödinger equation (NLSE). Our method relies on a compact ‘double-differencing’ which is shown to be computationally equivalent to standard fourth-order non-compact schemes. Through numerical simulations of the NLSE using fourth-order Runge-Kutta, we confirm that our scheme shows the desired fourth-order accuracy. A computation and storage requirement comparison is made between the 2SHOC scheme and the non-compact equivalent scheme for both the Laplacian operator alone, as well as when implemented in the NLSE simulations. Stability bounds are also shown in order to get maximum efficiency out of the method. We conclude that the modest increase in storage and computation of the 2SHOC schemes are well worth the advantages of having the schemes compact, and their ease of implementation makes their use very useful for practical implementations.

Key Words: High-order Compact Scheme, Nonlinear Schrödinger equation.

1 Introduction

The nonlinear Schrödinger equation (NLSE) is used to model a variety of physical systems [1] including Bose-Einstein condensates [2] and light propagation in nonlinear optical media [3]. This is because it describes, to lowest nonlinear order, modulated wave propagation. The general form of the NLSE can be written as

$$i\frac{\partial\Psi}{\partial t} + a\nabla^2\Psi - V(\mathbf{r})\Psi + s|\Psi|^2\Psi = 0, \quad (1)$$

where $\Psi(\mathbf{r}, t) \in \mathbb{C}$ is the value of the wavefunction, ∇^2 is the Laplacian operator, and where $a > 0$ and s are parameters defined by the system being modeled. $V(\mathbf{r})$ is an external potential term, which when included, makes Eq. (1) known as the Gross-Pitaevskii equation [2].

For almost every nonlinear partial differential equation (PDE) (including the NLSE), the interesting solutions require simulations using numerical methods. A standard methodology includes discretizing the solution in space by using finite-difference approximation schemes for the required derivatives. A *high-order* scheme is one which exhibits greater than second-order ($O(h^2)$, where h is the grid spacing) accuracy. While there are a large number of scenarios where higher-order schemes are a necessity due to the desired accuracy

*Corresponding author, email: sumseq@gmail.com, phone: 510-673-0720

[†]URL: <http://nlds.sdsu.edu>

[‡]URL: <http://www.csrc.sdsu.edu>

of the simulations, often the higher accuracy of high-order schemes is unnecessary, and second-order accuracy is sufficient for the problem. However, in such a case, using high-order schemes is still desirable because they allow one to simulate a solution with many fewer grid points, while maintaining the same accuracy as a second-order scheme. In those situations, the desired grid point size is based on the ability to resolve the structure of the solution, and not on the accuracy of computation.

Higher-order finite-difference schemes are typically achieved by computing derivatives with a wider scheme stencil. This adds a difficulty near the boundary, as one must be able to calculate the inner point near the boundary with the same accuracy as the internal scheme which can be complicated to implement. If this is not done, the entire simulation can eventually become lower-order. Another disadvantage of wide-stencils is that they cause many parallel implementations to be more difficult to realize, as different compute nodes must share or transfer more boundary values to their neighboring nodes. Besides the added complexity of the codes, the extra communication/global-memory-access (in the case of graphical processing units) reduces the overall parallel performance [4].

For the aforementioned reasons, there is great interest in high-order compact (HOC) schemes. These are finite-difference schemes which exhibit higher-order accuracy but still only rely on the closest neighboring points for computations. HOC schemes have been developed for various multidimensional steady-state PDEs [5–7], as well as time-dependent PDEs [8–12] including the one-dimensional NLSE [13, 14]. One drawback of HOC schemes are that their formulations are typically specific for the model equation being used, and requires re-deriving the schemes for each different PDE to be simulated (see Ref. [7], where the authors developed a Maple program to symbolically generate a HOC scheme for three-dimensional linear elliptical models).

The time-dependent HOC schemes developed so far are all implicit (even if the time-stepping is done with an otherwise explicit scheme), requiring solving a linear system in each time-step, as well as iterative processes when simulating nonlinear PDEs such as the NLSE (see however Ref. [13] where the author’s second implicit scheme implementation was able to be formulated to get around this requirement). The computational and storage requirements for implicit schemes can be prohibitive in large multidimensional settings. They are also, in general, difficult to optimally parallelize, especially in higher dimensions.

For these reasons, we wish to develop HOC formulations that are *fully explicit* time-dependent schemes (in our case, for the NLSE). To do this, we formulate a two-step procedure in computing the spatial derivatives, where each individual step is a compact computation. The first step of the 2SHOC scheme computes the standard second-order finite-difference approximation to the derivatives of the Laplacian, while the second step uses these computed derivative values to compactly approximate the Laplacian to fourth-order accuracy. Since the 2SHOC scheme computes the Laplacian independent of the governing PDE, it can be used in a multitude of time-dependent and time-independent PDEs which contain the Laplacian operator, increasing its generality. An explicit 2SHOC scheme for the NLSE can be formed by utilizing standard explicit ordinary differential equation (ODE) solvers. For the scheme described in this paper, we use the classic fourth-order Runge-Kutta (RK4) [15].

The paper is organized as follows. In Section 2 we show the formulation of the 2SHOC scheme for the Laplacian operator in one, two and three dimensions. Then, in Section 3, we use the scheme to form an explicit method for simulating the NLSE. In Section 4, we show numerical tests of each scheme from Section 3 and confirm their accuracy. In Section 5 we compare the storage and computation requirements for the 2SHOC schemes versus the standard fourth-order non-compact equivalent schemes. We conclude in Section 6.

2 Formulation of 2SHOC Schemes for the Laplacian Operator

It is well known that one can derive a fourth-order accurate finite-difference scheme for the second spatial derivative of a function by applying a ‘double-differencing’ approach. This works by noting that the second-order truncation term in the standard central-difference scheme contains the fourth spatial derivative. A second-order approximation to the fourth derivative can be obtained by applying a central-difference operator to the result of applying a central-difference operator to the function. When multiplied by the h^2 in the truncation term, the error in the truncation term becomes $O(h^4)$. Therefore, the resulting scheme becomes a fourth-order accurate approximation to the second derivative. When these two steps are combined and

simplified, one yields the standard non-compact fourth-order stencil. Due to the increase in the number of computations and storage, the ‘double-differencing’ procedure is not actually implemented, but rather the resulting non-compact stencil is used directly. However due to the advantages of using a compact scheme, this ‘double-differencing’ procedure becomes an overlooked, viable alternative to other, more complicated HOC schemes. In time-dependent systems, the sequential nature of the ‘double-differencing’ procedure makes it only viable for explicit time-stepping schemes, and since most time-stepping schemes in use today are implicit, this procedure has been overlooked.

2.1 One-Dimensional Formulation

Although the methodology for the 2SHOC scheme is to use the ‘double-differencing’ approach, it is worthwhile to show that the scheme can also be formed as a result of attempting to make other HOC schemes explicitly time-dependent. In Ref. [12], the authors formulate an implicit time-dependent HOC scheme for the transient convection-diffusion equation using their method from Ref. [6]. We can obtain a compact high-order scheme for the time-dependent NLSE following their methodology, and with a slight variation, can transform it into an explicit scheme which is equivalent to the 2SHOC.

In one dimension, we discretize the wavefunction of the NLSE as $\Psi(x_i, t_n) \equiv \Psi_i^n$ with a grid spacing of size h ($x_i = x_{\min} + hi$) and a time-step of size k ($t_n = kn$). The Laplacian operator applied to Ψ in one dimension can be represented as

$$\nabla^2 \Psi_i = \left. \frac{\partial^2 \Psi}{\partial x^2} \right|_i = \delta_x^2 \Psi_i - \frac{h^2}{12} \left(\left. \frac{\partial^4 \Psi}{\partial x^4} \right|_i \right) + O(h^4), \quad (2)$$

where the central-difference operator δ_x^2 is defined as

$$\delta_x^2 \Psi_i = \frac{1}{h^2} (\Psi_{i+1} - 2\Psi_i + \Psi_{i-1}). \quad (3)$$

As per Ref. [12], to formulate a high-order compact scheme, Eq. (1) is differentiated to form an expression for the fourth derivative in Eq. (2) which yields

$$\left. \frac{\partial^4 \Psi}{\partial x^4} \right|_i = -\frac{1}{a} \delta_x^2 \left[i \left. \frac{\partial \Psi}{\partial t} \right|_i + (s|\Psi_i|^2 - V(x_i)) \Psi_i \right] + O(h^2). \quad (4)$$

Due to the h^2 in the truncation term of Eq. (2), when Eq. (4) is inserted into Eq. (2), the resulting approximation of the right-hand-side of the NLSE becomes $O(h^4)$.

This method has been used in numerous publications for different governing equations [7, 9], however in each case the result is an implicit time-stepping scheme (even if the time derivatives are approximated by explicit differencing) due to the central-difference operator being applied to the time-derivative of Ψ in Eq. (4).

In order to retain a fully explicit scheme, one can use a two-step approach. First, an $O(h^2)$ approximation of $\left. \frac{\partial \Psi}{\partial t} \right|_i$ is evaluated by applying standard second-order central differencing on Eq. (1)

$$T_i = i \left[a \delta_x^2 \Psi_i + (s|\Psi_i|^2 - V(x_i)) \Psi_i \right] \approx \left. \frac{\partial \Psi}{\partial t} \right|_i. \quad (5)$$

Once computed, T_i is inserted into Eq. (4), and the spatial derivatives are then computed with central differencing. Since the error in evaluating Eq. (5) is $O(h^2)$, it too will result in an $O(h^4)$ error when Eq. (4) is inserted into Eq. (2). The boundary conditions for Eq. (5) must also be computed to at least $O(h^2)$ accuracy since the boundary points will be used for interior calculations of the spatial derivative of Eq. (5) (see Sec. 3.2 for details on boundary conditions for the 2SHOC schemes). Once Eq. (5) is computed with its boundary conditions, the second stage of the scheme is to then solve the semi-discrete equation

$$\left. \frac{\partial \Psi}{\partial t} \right|_i = i \left[a \left(\delta_x^2 \Psi_i + \frac{h^2}{a 12} \delta_x^2 \left[i T_i + (s|\Psi_i|^2 - V(x_i)) \Psi_i \right] \right) + (s|\Psi_i|^2 - V(x_i)) \Psi_i \right] + O(h^4). \quad (6)$$

One can use any desired explicit ODE solver (that is stable for the problem with its parameters) to compute the evolution of Eq. (6).

If one algebraically combines the two stages [Eq. (5) and Eq. (6) together, one gets

$$\frac{\partial \Psi}{\partial t} \Big|_i = i \left[a \left(\delta_x^2 \Psi_i - \frac{h^2}{12} \delta_x^2 (\delta_x^2 \Psi_i) \right) + (s |\Psi_i|^2 - V(x_i)) \Psi_i \right] + O(h^4), \quad (7)$$

and therefore the two-step HOC scheme described in Eq. (5) and Eq. (6) is computationally equivalent to simply taking the central-difference of the central-difference to approximate the fourth derivative truncation term in the Laplacian. Thus we recover the 2SHOC scheme approach for approximating the Laplacian which does not depend on the other terms of the governing equation. After collecting terms and simplifying the 2SHOC ‘double-differencing’, the scheme is given by

$$1) \quad D_i = \frac{1}{h^2} (\Psi_{i+1} - 2\Psi_i + \Psi_{i-1}), \quad (8)$$

$$2) \quad \nabla^2 \Psi_i \approx \frac{7}{6} D_i - \frac{1}{12} (D_{i+1} + D_{i-1}). \quad (9)$$

The implementation of boundary conditions for Eqs. (8) and (9) is discussed in Sec. 3.2.

When the two steps of Eqs. (8) and (9) are combined algebraically and simplified, the standard five-point non-compact fourth-order finite-difference approximation is recovered:

$$\nabla^2 \Psi_i = \frac{\partial^2 \Psi}{\partial x^2} \Big|_i = \frac{-\Psi_{i+2} + 16\Psi_{i+1} - 30\Psi_i + 16\Psi_{i-1} - \Psi_{i-2}}{12h^2} + O(h^4).$$

A potential drawback in using the 2SHOC scheme is that it requires extra storage space (the D vector) and more computations than using the standard fourth-order five-point stencil. However, as will be discussed in Sec. 5, the compact scheme’s advantages outweigh this deficiency.

2.2 Two-Dimensional Formulation

In two dimensions, the Laplacian operator applied to Ψ can be represented as

$$\nabla^2 \Psi_{i,j} = \frac{\partial^2 \Psi}{\partial x^2} \Big|_{i,j} + \frac{\partial^2 \Psi}{\partial y^2} \Big|_{i,j} = \delta_x^2 \Psi_{i,j} + \delta_y^2 \Psi_{i,j} - \frac{h^2}{12} \left(\frac{\partial^4 \Psi}{\partial x^4} \Big|_{i,j} + \frac{\partial^4 \Psi}{\partial y^4} \Big|_{i,j} \right) + O(h^4). \quad (10)$$

Once again, ‘double-differencing’ is used to obtain a second-order accurate approximation for the truncation term’s fourth derivatives. However, in this case, two different variations of the 2SHOC scheme can be formulated.

The first is a direct parallel to the one-dimensional 2SHOC scheme. Two matrices are formed, one which stores the second-order approximation to the second derivative in the x direction, and the other in the y direction. These are then used to form second-order approximations to the fourth derivatives in the truncation term in Eq. (10). After simplification, this results in the 2SHOC scheme

$$1) \quad D_{i,j}^x = \delta_x^2 \Psi_{i,j} = \frac{\Psi_{i+1,j} - 2\Psi_{i,j} + \Psi_{i-1,j}}{h^2}, \quad (11)$$

$$D_{i,j}^y = \delta_y^2 \Psi_{i,j} = \frac{\Psi_{i,j+1} - 2\Psi_{i,j} + \Psi_{i,j-1}}{h^2},$$

$$2) \quad \nabla^2 \Psi_{i,j} \approx \frac{7}{6} (D_{i,j}^x + D_{i,j}^y) - \frac{1}{12} (D_{i+1,j}^x + D_{i-1,j}^x + D_{i,j+1}^y + D_{i,j-1}^y). \quad (12)$$

The 2SHOC as described requires two storage matrices (D_x and D_y) in addition to that for Ψ . In large-scale computations, memory use can become a bottleneck and so it is useful to limit the amount of extra storage required as much as possible. We find that the two-dimensional 2SHOC scheme can be reformulated to only require one extra storage matrix, at the cost of requiring more computations (a trade-off that is evaluated in Sec. 5).

To formulate the lower-storage version of the two-dimensional 2SHOC, the standard five-point second-order finite-difference approximation to the two-dimensional Laplacian is used for the first stage (stored in D), and then the result is combined with a second-order cross-derivative stencil to yield the second-order approximations to the fourth derivative truncation terms in Eq. (10).

First, we note that

$$\nabla^2 (\nabla^2 \Psi) = \frac{\partial^2}{\partial x^2} \nabla^2 \Psi + \frac{\partial^2}{\partial y^2} \nabla^2 \Psi = \frac{\partial^4 \Psi}{\partial x^4} + \frac{\partial^4 \Psi}{\partial y^4} + 2 \frac{\partial^4 \Psi}{\partial x^2 \partial y^2},$$

in which case the fourth derivative truncation terms in Eq. (10) can be written as

$$\frac{\partial^4 \Psi}{\partial x^4} + \frac{\partial^4 \Psi}{\partial y^4} = \frac{\partial^2}{\partial x^2} \nabla^2 \Psi + \frac{\partial^2}{\partial y^2} \nabla^2 \Psi - 2 \frac{\partial^4 \Psi}{\partial x^2 \partial y^2}. \quad (13)$$

The cross-derivative in Eq. (13) is known to have the nine-point second-order compact stencil [16]

$$\left. \frac{\partial^4 \Psi}{\partial x^2 \partial y^2} \right|_{i,j} = \frac{1}{h^4} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \Psi_{i,j} + O(h^2). \quad (14)$$

Part of the stencil of Eq. (14) can be written in terms of the five-point second-order approximation to the Laplacian stored in D , in which case the single-storage 2SHOC scheme (after simplification) is written as:

$$1) \quad D_{i,j} = \delta_x^2 \Psi_{i,j} + \delta_y^2 \Psi_{i,j} = \frac{1}{h^2} \begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} \Psi_{i,j} \quad (15)$$

$$2) \quad \nabla^2 \Psi_{i,j} \approx -\frac{1}{12} \begin{bmatrix} & 1 & \\ 1 & -12 & 1 \\ & 1 & \end{bmatrix} D_{i,j} + \frac{1}{6h^2} \begin{bmatrix} 1 & & 1 \\ & -4 & \\ 1 & & 1 \end{bmatrix} \Psi_{i,j}. \quad (16)$$

As in the one-dimensional case, both formulations of the 2SHOC schemes in two dimensions are computationally equivalent to the non-compact fourth-order stencil

$$\nabla^2 \Psi_{i,j} = -\frac{1}{12h^2} \begin{bmatrix} & & 1 & & \\ & & -16 & & \\ 1 & -16 & 60 & -16 & 1 \\ & & -16 & & \\ & & 1 & & \end{bmatrix} \Psi_{i,j} + O(h^4). \quad (17)$$

2.3 Three-Dimensional Formulation

In three dimensions, the Laplacian operator applied to Ψ can be represented as

$$\begin{aligned} \nabla^2 \Psi_{i,j,k} &= \left. \frac{\partial^2 \Psi}{\partial x^2} \right|_{i,j,k} + \left. \frac{\partial^2 \Psi}{\partial y^2} \right|_{i,j,k} + \left. \frac{\partial^2 \Psi}{\partial z^2} \right|_{i,j,k} \\ &= \delta_x^2 \Psi_{i,j,k} + \delta_y^2 \Psi_{i,j,k} + \delta_z^2 \Psi_{i,j,k} - \frac{h^2}{12} \left(\left. \frac{\partial^4 \Psi}{\partial x^4} \right|_{i,j,k} + \left. \frac{\partial^4 \Psi}{\partial y^4} \right|_{i,j,k} + \left. \frac{\partial^4 \Psi}{\partial y^4} \right|_{i,j,k} \right) + O(h^4). \end{aligned} \quad (18)$$

There are again two formulations of the 2SHOC scheme which trade off storage requirement versus number of computations. The first formulation takes *three* storage matrices (D^x, D^y, D^z), while, as in the two-dimensional case, the other formulation takes only *one* (D). The three-storage 2SHOC directly follows from

the two-dimensional version and is defined as

$$1) \quad D_{i,j,k}^x = \delta_x^2 \Psi_{i,j,k} = \frac{\Psi_{i+1,j,k} - 2\Psi_{i,j,k} + \Psi_{i-1,j,k}}{h^2} \quad (19)$$

$$D_{i,j,k}^y = \delta_y^2 \Psi_{i,j,k} = \frac{\Psi_{i,j+1,k} - 2\Psi_{i,j,k} + \Psi_{i,j-1,k}}{h^2}$$

$$D_{i,j,k}^z = \delta_z^2 \Psi_{i,j,k} = \frac{\Psi_{i,j,k+1} - 2\Psi_{i,j,k} + \Psi_{i,j,k-1}}{h^2}$$

$$2) \quad \nabla^2 \Psi_{i,j,k} \approx \frac{7}{6} \left(D_{i,j,k}^x + D_{i,j,k}^y + D_{i,j,k}^z \right) - \frac{1}{12} \left(D_{i+1,j,k}^x + D_{i-1,j,k}^x + D_{i,j+1,k}^y + D_{i,j-1,k}^y + D_{i,j,k+1}^z + D_{i,j,k-1}^z \right). \quad (20)$$

The single storage formulation requires the use of the cross-derivatives of $\nabla^2(\nabla^2\Psi)$ as before. We write the second-order approximation of the Laplacian's fourth derivative truncation terms as

$$\frac{\partial^4 \Psi}{\partial x^4} + \frac{\partial^4 \Psi}{\partial y^4} + \frac{\partial^4 \Psi}{\partial z^4} = \nabla^2 (\nabla^2 \Psi) - 2 \left(\frac{\partial^4 \Psi}{\partial x^2 \partial y^2} + \frac{\partial^4 \Psi}{\partial x^2 \partial z^2} + \frac{\partial^4 \Psi}{\partial y^2 \partial z^2} \right). \quad (21)$$

The cross-derivatives in Eq. (21) can be approximated to second-order accuracy using nine-point compact stencils in each direction with the same coefficients as the two-dimensional stencil of Eq. (14).

After extracting D terms out of the cross derivatives and simplifying, the single-storage 2SHOC scheme in three dimensions becomes

$$1) \quad D_{i,j,k} = \frac{1}{h^2} \left(\begin{array}{|c|c|c|} \hline & & \\ \hline & 1 & \\ \hline & & \\ \hline \end{array} \Psi_{i,j+1,k} + \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & -6 & 1 \\ \hline & 1 & \\ \hline \end{array} \Psi_{i,j,k} + \begin{array}{|c|c|c|} \hline & & \\ \hline & 1 & \\ \hline & & \\ \hline \end{array} \Psi_{i,j-1,k} \right), \quad (22)$$

$$2) \quad \nabla^2 \Psi_{i,j,k} \approx -\frac{1}{12} \left(\begin{array}{|c|c|c|} \hline & & \\ \hline & 1 & \\ \hline & & \\ \hline \end{array} D_{i,j+1,k} + \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & -10 & 1 \\ \hline & 1 & \\ \hline \end{array} D_{i,j,k} + \begin{array}{|c|c|c|} \hline & & \\ \hline & 1 & \\ \hline & & \\ \hline \end{array} D_{i,j-1,k} \right) + \frac{1}{6h^2} \left(\begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & & 1 \\ \hline & 1 & \\ \hline \end{array} \Psi_{i,j+1,k} + \begin{array}{|c|c|c|} \hline 1 & & 1 \\ \hline & -12 & \\ \hline 1 & & 1 \\ \hline \end{array} \Psi_{i,j,k} + \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & & 1 \\ \hline & 1 & \\ \hline \end{array} \Psi_{i,j-1,k} \right). \quad (23)$$

Again, it is easy to show that both formulations of the 2SHOC scheme in three dimensions are computationally equivalent to the standard non-compact fourth-order stencil.

3 Implementation of the 2SHOC Schemes for solving the NLSE

In this section, we show the implementation of the 2SHOC schemes into explicit methods for simulating the NLSE. The NLSE is a good model for use with the 2SHOC schemes, as it is very relevant in current research, and its nonlinear terms make the use of explicit time-stepping desirable.

3.1 Explicit Time Integration using Runge-Kutta

To formulate a fully explicit scheme for simulating the NLSE, a method of lines approach is utilized, where the 2SHOC is used for the spatial Laplacian and the resulting semi-discrete system of ODEs is simulated using an explicit time-stepping scheme. As shown in Ref. [17], the first-order forward difference and the second-order Runge-Kutta scheme (Heun's method) for simulating the NLSE are unconditional unstable. Thus a higher-order multi-step method is needed, so we choose to use the standard fourth-order Runge-Kutta (RK4) scheme [15]. To facilitate our computational cost comparisons in Sec. 5, we write the RK4

scheme algorithmically as

$$\begin{aligned}
1) K_{\text{tot}} &= F(\Psi^n) & 6) K_{\text{tmp}} &= F(\Psi_{\text{tmp}}) \\
2) \Psi_{\text{tmp}} &= \Psi^n + \frac{k}{2} K_{\text{tot}} & 7) K_{\text{tot}} &= K_{\text{tot}} + 2 K_{\text{tmp}} \\
3) K_{\text{tmp}} &= F(\Psi_{\text{tmp}}) & 8) \Psi_{\text{tmp}} &= \Psi^n + k K_{\text{tmp}} \\
4) K_{\text{tot}} &= K_{\text{tot}} + 2 K_{\text{tmp}} & 9) K_{\text{tmp}} &= F(\Psi_{\text{tmp}}) \\
5) \Psi_{\text{tmp}} &= \Psi^n + \frac{k}{2} K_{\text{tmp}} & 10) \Psi^{n+1} &= \Psi^n + \frac{k}{6} (K_{\text{tot}} + K_{\text{tmp}}),
\end{aligned} \tag{24}$$

where

$$F(\Psi) = \frac{\partial \Psi}{\partial t} = i [a \nabla^2 \Psi + (s |\Psi|^2 - V(\mathbf{r})) \Psi],$$

and the two steps of the 2SHOC scheme are used to evaluate $\nabla^2 \Psi$ inside $F(\Psi)$. We denote the combined time-dependent scheme as RK4+2SHOC.

The RK4 time-stepping requires three storage matrices ($K_{\text{tmp}}, K_{\text{tot}}, \Psi_{\text{tmp}}$) in addition to the storage for the solution Ψ and external potential $V(\mathbf{r})$ (if $V(\mathbf{r})$ is chosen to be stored rather than computed at each evaluation). Lower-storage fourth-order Runge-Kutta schemes with comparable accuracy have been developed (a $3N$ in Ref. [18], and a five-stage $2N$ in Ref. [19] which requires an additional function evaluation, and whose coefficients are numerically derived), but for simplicity we use the classic RK4 of Eq. (24). Using the 2SHOC scheme inside $F(\Psi)$ requires additional storage, which we discuss in Sec. 5.

The RK4+2SHOC scheme is conditionally stable, in that the size of the time-step is limited by a bound based on the spatial step h (see Sec. 3.3 for the stability bounds of the RK4+2SHOC scheme applied to the NLSE). Conditional stability is one of the only drawbacks of using explicit schemes. However, even though implicit schemes are usually unconditionally stable, error requirements and algorithm complexity often make the explicit schemes more efficient even taking the time-step limitations into account.

3.2 Boundary Conditions

The use of proper boundary conditions is very important when performing numerical simulations. The 2SHOC scheme requires two boundary conditions, one for each step. The first step is a boundary condition on the Laplacian of the wavefunction (or on the separate spatial derivatives if the high-storage 2SHOC is being used). The second step either requires a boundary condition for the fourth spatial derivatives, or in many cases, the scheme for the overall model equations dictate the boundary condition (for example, in our RK4 scheme for the NLSE, the boundary condition of the second step of the 2SHOC is not required, as the boundary condition of the time derivative of the wavefunction (Ψ_t) overwrites any condition that would be in the 2SHOC scheme).

As mentioned in Ref. [12], it appears that in general, compact boundary conditions are not possible to realize for one-stage HOC schemes. However, even when the use of one-sided differencing (or any other non-compact technique) is necessary, it does not negate the boundary advantage of using HOC schemes (since the advantage is not having to alter the scheme *near* the boundaries). There are some boundary condition techniques that can be seen as compact, and hence fit very well into the framework of HOC schemes. These conditions are a Dirichlet ($\Psi = \text{const}$) and modulus-squared Dirichlet (MSD) ($|\Psi|^2 = \text{const}$) [20] boundary conditions.

In many scenarios, one would ideally want a transparent boundary condition. However, such conditions can be complicated to implement (see Ref. [21] for a review of different approaches). For many problems, an easy alternative is to make the computational grid large enough so that one can reasonably use Dirichlet conditions at the boundaries. When $s > 0$ (attractive or focusing nonlinearity) in the NLSE, most dynamics are of areas of density which decay into a surrounding background area of zero-density, in which case a Dirichlet boundary condition of zero is useful. In order to use the Dirichlet boundary condition with the 2SHOC scheme, the boundary condition on the Laplacian must be computed in the first step. This is easily found by setting $\Psi_t = 0$ in Eq. (1) yielding

$$\nabla^2 \Psi_b = -\frac{(s |\Psi_b|^2 - V_b) \Psi_b}{a}, \tag{25}$$

where b represents a boundary grid point. In the second stage of the 2SHOC, the boundary condition of Eq. (25) would also be used for the overall Laplacian boundary. In the RK4+2SHOC method for the NLSE, the second stage boundary condition would simply apply the Dirichlet condition directly as

$$\left. \frac{\partial \Psi}{\partial t} \right|_b = 0. \quad (26)$$

In the case where $s < 0$ (repulsive or defocusing nonlinearity) and $V(\mathbf{r}) = 0$ in the NLSE, most interesting dynamics are within areas of low densities which increase towards an area with a constant background density. In such a case, one cannot use standard Dirichlet boundary conditions since the constant boundary is in the modulus-squared of the wavefunction, not at a single real and imaginary value. To solve this problem, we recently have developed a new modulus-squared Dirichlet boundary condition that accurately simulates a constant density at the boundaries [20]. This boundary condition is described as

$$\left. \frac{\partial \Psi}{\partial t} \right|_b \approx \left. \frac{\partial \Psi}{\partial t} \right|_{b-1} \frac{\Psi_b}{\Psi_{b-1}}, \quad (27)$$

where $b - 1$ represents the closest internal point relative to a boundary grid point in the normal direction. The standard form of the MSD boundary condition requires to first compute all interior points and then compute the boundaries, making the general form of the MSD not compatible with implicit schemes (see Ref. [20] for details on possible implicit scheme implementation strategies of the MSD).

The MSD boundary condition can be used for any time-dependent complex PDE and in multiple dimensions. For our application to the NLSE, we require boundary conditions for each step of the 2SHOC scheme. For the first step, one can substitute the NLSE in the MSD boundary condition and get an MSD condition for the Laplacian given by

$$\nabla^2 \Psi_b \approx \left[\frac{\nabla^2 \Psi_{b-1}}{\Psi_{b-1}} + \frac{1}{a} (V_b - V_{b-1} + s (|\Psi_{b-1}|^2 - |\Psi_b|^2)) \right] \Psi_b, \quad (28)$$

while for the second step, the boundary condition of Eq. (27) can (in the RK4+2SHOC scheme) be used directly.

It should be noted that the MSD boundary condition cannot be directly applied to the multi-storage version of the 2SHOC schemes because the MSD computes the boundary value of $\nabla^2 \Psi$, not the individual second spatial derivatives needed for Eq. (11) or Eq. (19). This can be overcome by rearranging the boundary condition taking advantage of the cross derivative stencils. However, since we will be only using the single-storage 2SHOC schemes in our examples, we do not show this alteration to the MSD boundary condition here.

There are numerous other boundary conditions one might use for the NLSE including other compact-friendly ones such as Laplacian-zero ($\nabla^2 \Psi = 0$) and periodic. However, in this paper we have limited ourselves to Dirichlet and MSD, as they will be used in the numerical tests of Sec. 4.

3.3 Stability

Since the most important drawback in using explicit schemes is that they are conditionally stable, it is very important to limit this deficiency as much as possible by computing the stability bound to determine the largest time-step that is usable.

Since the 2SHOC schemes are algebraically equivalent to the standard non-compact fourth-order schemes, the stability bounds on the time-step should not adversely be affected by the use of the compact schemes. However, the form of the boundary condition implementation in the 2SHOC schemes will change the way the near-boundary points are computed. In Ref. [17] we used an extension to the methodology employed in Ref. [22] to conduct an analysis of the stability bounds for simulating the multidimensional NLSE with RK4 for various boundary conditions including those in Sec. 3.2. The analysis was done for both the standard second-order central differencing, as well as fourth-order differencing using the 2SHOC formulation. The results utilizing the two boundary conditions discussed in Sec. 3.2 are summarized as follows:

In the linear case where $s = 0$ and with no external potential ($V(\mathbf{r}) = 0$), utilizing Dirichlet boundary conditions, the stability bound on the time-step k using a fourth-order central-difference scheme (with interior points computed in the 2SHOC methodology) in a d -dimensional setting is

$$k_{\text{linear}} < \left(\frac{3}{4}\right) \frac{h^2}{d\sqrt{2}a}, \quad (29)$$

which is just 3/4 of the stability bound in the second-order central differencing case.

In the general NLSE case, one can get a linearized stability bound by treating the nonlinearity $|\Psi|^2$ (and the MSD boundary condition terms) as a constant, yielding

$$k < \frac{\sqrt{8}}{\max\{\|\vec{B}\|_\infty, \|\nabla L_i, L_i - \vec{G}\|_\infty\}} \frac{h^2}{a}, \quad (30)$$

where \vec{B} are the boundary points as defined by Table 1, \vec{L} is defined as

$$L_i = \frac{h^2}{a} (s|\Psi_i|^2 - V(\mathbf{r}_i)),$$

and \vec{G} is a set of values defined in Table 2, determined by the dimension being used.

Table 1: Values for B_b in Eq. (30).

	Dirichlet ($\Psi_b = \text{const}$)	MSD ($ \Psi_b ^2 = \text{const}$)
B_b	0	$\frac{h^2}{i a} \frac{1}{\Psi_{b-1}} \frac{\partial \Psi}{\partial t} \Big _{b-1}$

Table 2: Values for \vec{G} in Eq. (30) for the one-, two-, and three-dimensional NLSE.

	$d = 1$	$d = 2$	$d = 3$
\vec{G}	$\frac{1}{12} \times \{64, 63, 46, 12, -3, -4\}$	$\frac{1}{12} \times \{128, 127, 126, 110, 109, 92, 24, 9, 8, -6, -7, -8\}$	$\frac{1}{12} \times \{192, 191, 190, 189, 174, 173, 172, 156, 155, 138, 36, 21, 20, 6, 5, 4, -9, -10, -11, -12\}$

Since these stability results are based on a linearized approximation to the full nonlinear problem, in practice, one must use a time-step that is lower than the bounds given. In our experience, setting the time-step to be 80% of the given bounds ensures stability in most cases (in one-dimensional simulations, the step size can be up to 90% of the given bounds for most cases).

4 Numerical Results

Here we show our numerical results for using the RK4+2SHOC scheme to integrate the NLSE. Since the 2SHOC schemes are algebraically equivalent to the standard fourth-order non-compact schemes, they should exhibit the same order of accuracy as well. However, since the order of computation is altered (which could introduce numeric cancellation or truncation errors), and the implementation of boundary conditions is different, we feel that numerical tests are justified. For each numerical test we use our GPU-accelerated code package NLSEmagic (see Ref. [23]) and pick non-trivial initial conditions. For all tests we utilize the single-storage formulation of the 2SHOC schemes.

4.1 One-Dimensional Test

For the one-dimensional test of the RK4+2SHOC scheme, we use a known dynamical solution to the one-dimensional NLSE (with $V(\mathbf{r}) = \mathbf{0}$ and $s < 0$) of a co-moving dark soliton given by [2]

$$\Psi(x, t) = \sqrt{\frac{\Omega}{s}} \tanh \left[\sqrt{\frac{|\Omega|}{2a}} (x - ct) \right] \exp \left(i \left[\frac{c}{2a} x + \left(\Omega - \frac{c^2}{4a} \right) t \right] \right), \quad (31)$$

where Ω is the frequency and c is the velocity of the soliton. We use the modulus-squared Dirichlet boundary condition [Eqs. (27) and (28)] and set the size of the computational domain large enough so that the wavefunction's density is machine epsilon ($\epsilon \approx 10^{-16}$) lower than the background density at the boundaries throughout the entire time of the simulation. We set $s = -1$, $a = 1$, $c = 0.5$, and $\Omega = -1$ in Eq. (31). The grid spatial-step h is varied from $1/2$ to $1/32$. We run the simulation to an end-time of $t = 10$ with a time-step size of $k = 0.0005$ (which is slightly less than the maximum stability bound for the smallest value of h used) resulting in 20,000 time steps. Since the RK4 scheme is $O(k^4)$, and the stability bounds require that $h \propto k^2$, the error in the simulations attributed to the time-stepping is negligible compared to the error due to the spatial differencing.

To compute the error in the simulations, the real and imaginary parts of the wavefunction Ψ are compared to the true solution of Eq. (31) at 100 equal-spaced intervals throughout the simulation. The averaged 2-norm of the wavefunction error ($E^n = \|\Psi - \Psi_{\text{exact}}\|_2^n / N$, where n is the current time step and N is the grid size) is computed at each time interval. (Using the averaged 2-norm error is necessary since we fix the domain size in x and, therefore, the total 2-norm will be greater for smaller h due to the increase in the grid size required). When the simulation is complete, we define the error of the real and imaginary parts of the whole simulation as the mean of the errors at each of the 100 intervals ($E = \Sigma E^n / K$, where $K = 100$ is the number of time intervals). To compute the order of error, we define the error order between two simulations with spatial steps h_1 and h_2 as $O = \ln(E_{h_1}) - \ln(E_{h_2})$. The overall order of the scheme is determined by taking the mean of the average of the real and imaginary error orders for each run. For comparison, we run each simulation using the classic second-order central-differencing (CD) in space. The results of the simulations are shown in Fig. 1. The fourth-order accuracy of the RK4+2SHOC scheme is easily observed.

4.2 Two-Dimensional Test

There is no readily available, non-trivial, exact two-dimensional solution to the NLSE to use for order comparisons. However, since the RK4+2SHOC scheme is explicit (and therefore does not require any iterative methods to implement the nonlinearity), the accuracy of the scheme can be tested reliably in a linear setting ($s = 0$). The chosen test problem is the exact (Gaussian wavepacket) solution

$$\Psi(x, y, t) = \exp \left(-\frac{x^2 + y^2}{2a} \right) \exp(-i 2t), \quad (32)$$

where $V(x, y)$ is the external potential

$$V(x, y) = \frac{x^2 + y^2}{a}. \quad (33)$$

We use Dirichlet boundary conditions ($\Psi = 0$) and set $a = 1$. We set the size of the computational domain large enough so that the exact solution has a value of $\Psi_b = \sqrt{\epsilon}$ (where $\epsilon \approx 10^{-16}$) at the boundaries. The simulation time and number of intervals for error computations are the same as in the one-dimensional test in Sec. 4.1. We vary h from $h = 1$ to $h = 1/16$ and use a time-step of $k = 0.001$ resulting in 10,000 time steps. The results are shown in Fig. 2. The fourth-order accuracy of the RK4+2SHOC scheme is once again observed.

4.3 Three-Dimensional Test

As in the two-dimensional case, there is no readily available non-trivial exact solution to test the three-dimensional NLSE. Therefore, we again focus on a linear example ($s = 0$) and use the three-dimensional

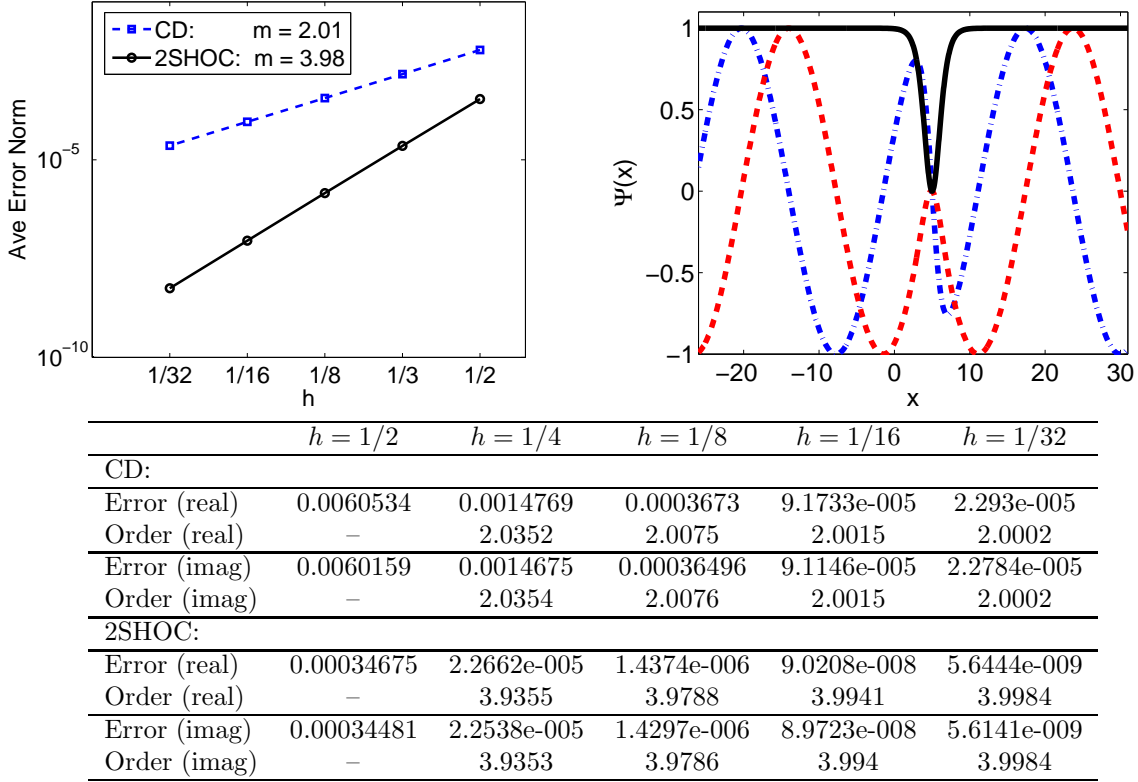
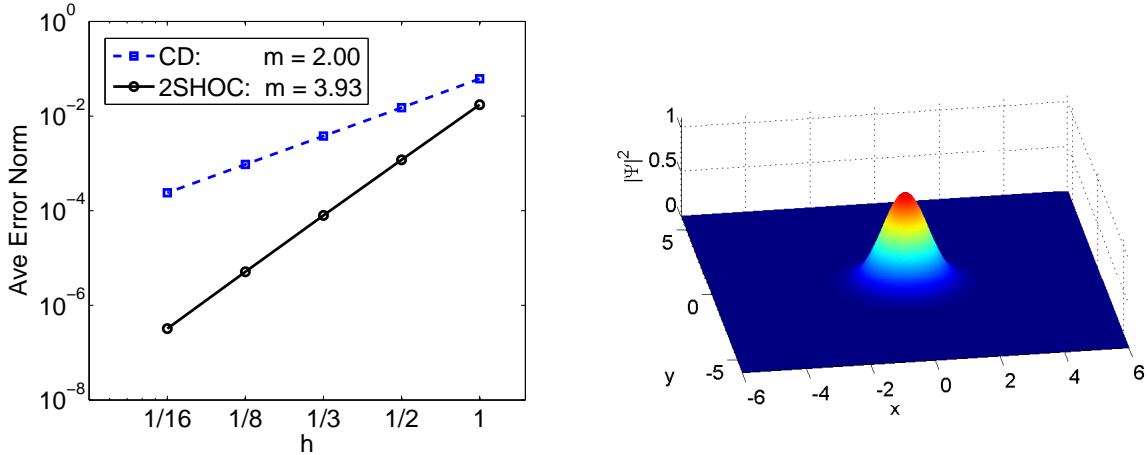


Figure 1: (Color online) Top Left: Overall order of accuracy (m) computed from simulating the one-dimensional NLSE with the initial condition of Eq. (31) for the RK4+CD and RK4+2SHOC schemes. Top Right: Snap-shot of the simulation with $h = 1/32$. The solid (black) line is the modulus-squared $|\Psi|^2$, while the dot-dashed (blue) and dashed (red) lines are the real and imaginary parts of Ψ respectively. Bottom: Table of error values and order of accuracy values for each spatial step h . The parameters used for the solution are $s = -1$, $a = 1$, $c = 0.5$, and $\Omega = -1$. The solution is integrated to an end-time of $t = 10$ with a time-step size of $k = 0.0005$.



	$h = 1$	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$
CD:					
Error (real)	0.059782	0.01451	0.0036626	0.00092345	0.00023196
Order (real)	-	2.0426	1.9861	1.9878	1.9932
Error (imag)	0.062855	0.015442	0.0038944	0.00097946	0.00024585
Order (imag)	-	2.0252	1.9874	1.9913	1.9942
2SHOC:					
Error (real)	0.017015	0.0011492	7.7072e-005	4.9388e-006	3.1147e-007
Order (real)	-	3.8882	3.8982	3.964	3.987
Error (imag)	0.017684	0.001228	8.2419e-005	5.2815e-006	3.3308e-007
Order (imag)	-	3.848	3.8972	3.964	3.987

Figure 2: Top Left: Overall order of accuracy (m) computed from simulating the two-dimensional linear Schrödinger equation with the initial condition of Eq. (32) for the RK4+CD and RK4+2SHOC schemes. Top Right: Depiction of the modulus-squared $|\Psi|^2$ of the wavefunction with $h = 1/16$ at $t = 0$. Bottom: Table of error values and order of accuracy values for each spatial step h . The parameters used for the solution are $s = 0$, and $a = 1$. The solution is integrated to an end time of $t = 10$ with a time-step size of $k = 0.001$.

analog to Eq. (32) defined as

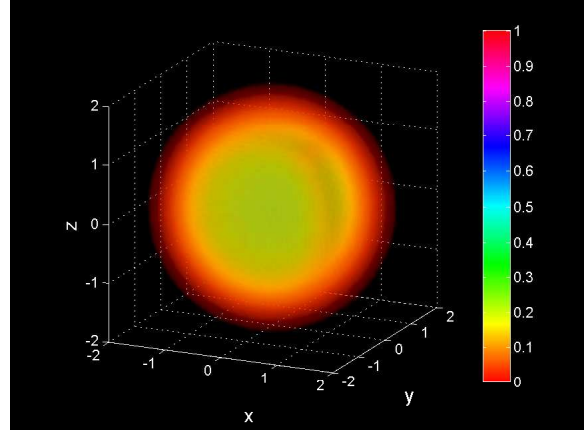
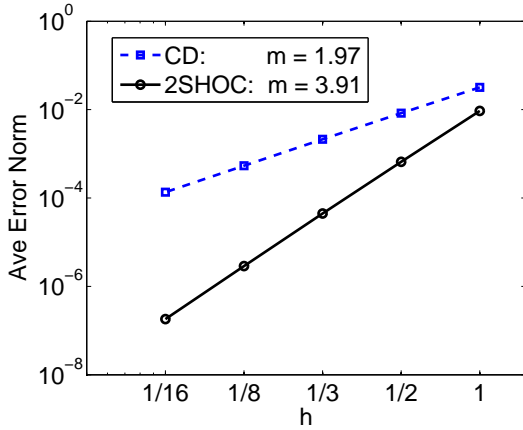
$$\Psi(x, y, t) = \exp\left(-\frac{x^2 + y^2 + z^2}{2a}\right) \exp(-i3t), \quad (34)$$

with external potential

$$V(x, y) = \frac{x^2 + y^2 + z^2}{a}. \quad (35)$$

We use Dirichlet boundary conditions ($\Psi = 0$) and set $a = 1$. We set the size of the computational domain large enough so that the exact solution has a value of $\Psi_b = \sqrt{\epsilon}$ at the boundaries. The simulation time and number of intervals for error computations are the same as in the one-dimensional test in Sec. 4.1. We vary h from $h = 1$ to $h = 1/16$ and use a time-step of $k = 0.0005$ resulting in 20,000 time steps. The results are shown in Fig. 3, where again the fourth-order accuracy of the scheme is seen. The average order of accuracy given in Fig. 3 is 3.91 which is smaller than expected for a fourth-order scheme. However as seen in the corresponding table, the order of accuracy starts at around 3.8 for the first spatial step-size reduction, but for the smaller reductions, the order increases to around 3.98. This means that the slightly lower-order values are due to the inability of the coarse grid to adequately resolve the solution, and not on the scheme itself. Lower values of h were not used due to computational memory constraints.

From the table in Fig. 3, it is easy to illustrate the advantage of using high-order schemes for large three-dimensional problems mentioned in Sec. 1, that of being able to use a much smaller grid while maintaining



	$h = 1$	$h = 1/2$	$h = 1/4$	$h = 1/8$	$h = 1/16$
CD:					
Error (real)	0.03125	0.0083706	0.0021518	0.00054571	0.00013743
Order (real)	–	1.9004	1.9598	1.9793	1.9894
Error (imag)	0.032541	0.0083009	0.0021026	0.00053144	0.00013374
Order (imag)	–	1.9709	1.9811	1.9842	1.9905
2SHOC:					
Error (real)	0.0093495	0.000666	4.5165e-005	2.9087e-006	1.839e-007
Order (real)	–	3.8113	3.8822	3.9567	3.9834
Error (imag)	0.0093489	0.00064941	4.4012e-005	2.8344e-006	1.792e-007
Order (imag)	–	3.8476	3.8832	3.9568	3.9834

Figure 3: Top Left: Overall order of accuracy (m) computed from simulating the three-dimensional linear Schrödinger equation with the initial condition of Eq. (34) for the RK4+CD and RK4+2SHOC schemes. Top Right: Volumetric rendering of the modulus-squared $|\Psi|^2$ of the wavefunction with $h = 1/16$ at $t = 0$ (displayed on a smaller grid than used in the simulations). Bottom: Table of error values and order of accuracy values for each spatial step h . The parameters used for the solution are $s = 0$, and $a = 1$. The solution is integrated to an end time of $t = 10$ with a time-step size of $k = 0.0005$.

the desired accuracy. For example, in Fig. 3, we see that using the second-order scheme for $h = 1/8$ (making the grid resolution $97 \times 97 \times 97 = 912,673$ grid points) yielded an error of around 0.0005. Roughly the same error (0.0006) was found using the fourth-order scheme with a spatial-step size of $h = 1/2$, requiring the grid size of only $25 \times 25 \times 25 = 15,625$ grid points. This is a 98.3% reduction in the number of grid points required! When combined with the ease of parallelization that the 2SHOC compact scheme provides, the usefulness of the 2SHOC scheme for large three-dimensional problems is prominent.

5 Computation and Storage Comparisons

In this section we compare the storage and computational requirements of the 2SHOC schemes compared to the standard fourth-order non-compact equivalent schemes for one, two and three dimensions. We count the number of operations (in terms of the number of elements in the domain, N) needed for each method ignoring the boundaries. For simplicity, we also ignore the added operations needed due to Ψ being complex, and treat all operations as acting on real variables. Also, since floating-point division operations are far more computationally expensive than additions and multiplications, we record the operations required in an optimized form of the schemes, in which all divisions are only computed once (and hence ignored) by pre-computing the constant terms. We record the number of computations and storage space required for the Laplacian operator alone using the 2SHOC, as well as when implemented into the NLSE simulations using the RK4+2SHOC schemes.

The one-dimensional analysis is shown in Table 3. We see that for the RK4+2SHOC NLSE implemen-

Table 3: One-dimensional storage and computational cost analysis for the 2SHOC scheme compared to the equivalent non-compact scheme. The storage is given in terms of N , the total number of grid points.

Method	Operations	Storage	Op Ratio	Storage Ratio
Laplacian:				
Non-Compact	7	N	–	–
2SHOC	8	$2N$	1.14	2
NLSE RK4 Step:				
Non-Compact	$4(7+7)+13 = 69$	$5N$	–	–
2SHOC	$4(8+7)+13 = 73$	$6N$	1.06	1.2

tation, the scheme only requires about 6% more computations and 20% more storage than the equivalent non-compact scheme. This small increase in storage and computations is minor compared to the advantages of having a compact scheme.

The two-dimensional analysis is given in Table 4. Both the double-storage version of the 2SHOC of Eqs. (11) and (12) and the single-storage version of Eqs. (15) and (16) are analyzed. Here, using the 2SHOC

Table 4: Two-dimensional storage and computational cost analysis for 2SHOC schemes compared to equivalent non-compact schemes. The storage is given in terms of N , the total number of grid points.

Method	Operations	Storage	Op Ratio	Storage Ratio
Laplacian:				
Non-Compact	9	N	–	–
2SHOC (2X Storage)	15	$3N$	1.66	3
2SHOC (1X Storage)	19	$2N$	2.11	2
NLSE RK4 Step:				
Non-Compact	$4(9+7) + 13 = 77$	$5N$	–	–
2SHOC (2X Storage)	$4(15+7)+13 = 101$	$7N$	1.31	1.4
2SHOC (1X Storage)	$4(19+7)+13 = 117$	$6N$	1.52	1.2

scheme is a greater increase in additional computations, taking 50% more (for the 1X storage version) than the non-compact scheme in the RK4 step of the NLSE. The two-storage version of 2SHOC reduces this to about 30%, but with a 20% increase in the storage required when compared to the single-storage 2SHOC (which like the one-dimensional case is only 20% more than the non-compact scheme).

The three-dimensional analysis is given in Table 5. Here we see that the single-storage 2SHOC scheme in the RK4 step for the NLSE takes 70% more computations than the non-compact scheme, and for the Laplacian operator alone, takes a little more than twice the computations required for the non-compact scheme (although it also requires four times the storage). However, since the main advantage of using a compact scheme is that it is easy to implement in a parallel environment, this increase in computation is easily offset by the parallelism (as most parallel implementations have speed-ups of well over two). Parallel implementations of the single-storage 2SHOC schemes have been written for use with graphical processing units (see Ref. [23]).

Table 5: Three-dimensional storage and computational cost analysis for 2SHOC schemes compared to equivalent non-compact schemes. The storage is given in terms of N , the total number of grid points.

Method	Operations	Storage	Op Ratio	Storage Ratio
Laplacian:				
Non-Compact	14	N	–	–
2SHOC (3X Storage)	22	$4N$	1.57	4
2SHOC (1X Storage)	31	$2N$	2.21	2
NLSE RK4 Step:				
Non-Compact	$4(14+7)+13 = 97$	$5N$	–	–
2SHOC (3X Storage)	$4(22+7)+13 = 129$	$8N$	1.33	1.6
2SHOC (1X Storage)	$4(31+7)+13 = 165$	$6N$	1.70	1.2

6 Conclusion

We have formulated two-step high-order compact (2SHOC) finite-difference schemes for the Laplacian operator in one, two, and three dimensions. Two forms of the multi-dimensional 2SHOC schemes were formulated, one requiring less storage and more computation than the other. When implementing the schemes in partial differential equation models, the schemes do not rely on other parts of the governing equation, and are therefore more general than other HOC schemes.

The schemes are well-suited for use in explicit finite-difference schemes for time-dependent problems. As an example, we have shown the implementation of the 2SHOC schemes into explicit schemes for solving the nonlinear Schrödinger equation and, through numerical simulations, have shown that the schemes display the desired accuracy.

A computation and storage analysis revealed that the 2SHOC schemes take more storage and computations (never more than roughly double) than their non-compact equivalent schemes, but that these increases are well offset due to the advantages of using a compact scheme, most notably, the easing of parallel implementations.

Acknowledgments

This research was supported by NSF-DMS-0806762 and the Computational Science Research Center at San Diego State University.

References

- [1] L. Debnath. *Nonlinear Partial Differential Equations for Scientists and Engineers*. Birkhauser Boston, New York, New York, 2 edition, 2005.
- [2] P. G. Kevrekidis, D. J. Frantzeskakis and R. Carretero-González. *Emergent Nonlinear Phenomena in Bose-Einstein Condensates: Theory and Experiment*, volume 45. Springer Series on Atomic, Optical, and Plasma Physics, 2008.
- [3] B. Malomed. Variational methods in nonlinear fiber-optics and related fields. *Progress in Optics*, **43** (2002) 71–193.
- [4] W. F. Spitz and G. F. Carey. Iterative and parallel performance of high-order compact systems. *SIAM Journal on Scientific Computing*, **19**, 1 (1998) 1–14.
- [5] J. Zhang. An explicit fourth-order compact finite difference scheme for three-dimensional convection-diffusion equation. *Communications in numerical methods in engineering*, **14** (1998) 209–218.

- [6] W. F. Spitz and G. F. Carey. A high-order compact formulation for the 3d poisson equation. *Numerical Methods for Partial Differential Equations*, **12** (1996) 235–243.
- [7] L. Ge and J. Zhang. Symbolic computation of high-order compact difference schemes for three dimensional linear elliptic partial differential equations with variable coefficients. *Journal of computational and applied mathematics*, **143** (2002) 9–27.
- [8] W. Liao. An implicit fourth-order compact finite difference scheme for one-dimensional burgers' equation. *Applied Mathematics and Computation*, **206** (2008) 755–764.
- [9] J. C. Kalita, P. Chhabra and S. Kumar. A semi-discrete higher order compact scheme for the unsteady two-dimensional Schrödinger equation. *Journal of computational and applied mathematics*, **197** (2006) 141–149.
- [10] A. Mohebbi and M. Dehghan. The use of compact boundary value method for the solution of two-dimensional Schrödinger equation. *Journal of computational and applied mathematics*, **225** (2009) 124–134.
- [11] M. Ciment and S. H. Leventhal. Higher order compact implicit schemes for the wave equation. *Mathematics of Computation*, **29**, 132 (1975) 985–994.
- [12] W. F. Spitz and G. F. Carey. Extension of high-order compact schemes to time-dependent problems. *Numerical Methods for Partial Differential Equations*, **17**, 6 (2001) 657–672.
- [13] S. Xie, G. Li and S. Yi. Compact finite-difference schemes with high accuracy for one-dimensional nonlinear Schrödinger equation. *Computational methods in applied mechanics and engineering*, **198** (2009) 1052–1060.
- [14] M. Dehghan and A. Taleei. A compact split-step finite difference method for solving the nonlinear Schrödinger equations with constant and variable coefficients. *Computer Physics Communications*, **181** (2010) 43–51.
- [15] G. H. Golub and J. M. Ortega. *Scientific Computing and Differential Equations An Introduction to Numerical Methods*. Academic Press, San Diego, California, 2 edition, 1992.
- [16] W. F. Spitz. *High-order compact finite difference schemes for computational mechanics*. PhD thesis, University of Texas at Austin, 1995.
- [17] R. M. Caplan and R. Carretero-González. Numerical stability of explicit runge-kutta finite differenceschemes for the nonlinear Schrödinger equation. *arXiv:1107.4810*, (2011).
- [18] D. J. Fyfe. Economical evaluation of runge-kutta formulae. *Math. Comp.*, **20** (1966) 392–398.
- [19] M. H. Carpenter and C. A. Kennedy. Fourth-order $2N$ -storage Runge-Kutta schemes. *NASA Technical Memorandum*, (1994).
- [20] R. M. Caplan and R. Carretero-González. A modulus-squared dirichlet boundary condition for time-dependant complex partial differential equations and its application to the nonlinear Schrödinger equation. *In Preparation*, (2011).
- [21] X. Antoine, A. Arnold, C. Besse, M. Ehrhardt and A. Schädle. A review of transparent and artificial boundary conditions techniques for linear and nonlinear Schrödinger equations. *Comm. in Comp. Phys.*, **4**, 4 (2008) 729–796.
- [22] M. S. Ismail. A fourth-order explicit schemes for the coupled nonlinear Schrödinger equation. *Applied Mathematics and Computation*, **196** (2008) 273–284.

- [23] R. M. Caplan. NLSEmagic: Nonlinear Schrödinger equation multidimensional Matlab-based GPU-accelerated integrators using compact high-order schemes. <http://www.nlsemagic.com>, (2011).